

## TITLE OF INVENTION

System and method for voice commands recognition and controlling devices wirelessly using protocol based communication

Inventor Name: Asim Hussain Abbasi

Citizenship: Pakistan

Residence: 110 Hidden Oaks Drive, Apt. 2C,  
Cary, North Carolina 27513, USA.

## CROSS-REFERENCE TO RELATED APPLICATIONS

Lots of work has been done in the field of voice recognition and numerous applications, systems have been developed so far based on this technology. I like to put a reference to the following patents that contains the idea of voice commands and control,

U.S. patent U.S. Pat. No. 6,188,985 entitled "Wireless voice-activated device for control of a processor-based host system"

U.S. patent U.S. Pat. No. 5,199,080 entitled "Voice-operated remote control system"

U.S. patent U.S. Pat. No. 4,704,696 entitled "Method and apparatus for voice control of a computer"

## STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not Applicable

## REFERENCE TO SEQUENCE LISTING, A TABLE OR A COMPUTER PROGRAM LISTING COMPACT DISK APPENDIX

Not Applicable

## BACKGROUND OF INVENTION

The present invention is related to using voice recognition engine, serial port interfacing, digital protocol writing, wireless communication, microcontroller programming and design of switching circuits.

Many different companies have developed voice recognition engines along with APIs (Application Programming Interface) to allow programmers having no knowledge about DSP

(Digital Signal Processing) or voice recognition to embed high quality voice recognition ability into their softwares using APIs and develop applications based on it.

Serial port interfacing describes how DTE (Data Terminal Equipment) communicates data serially using such ports on PC with DCE (Data Communication Equipment) and how to interface such ports with other electrical circuitry. Two serial ports are present in the form of DB-25 (called as Printer Port) and DB-9 in PC.

Digital protocol writing requires the knowledge of electrical as well as software engineering. By writing protocol we actually develop a Language using which different systems (software/hardware) communicates with each other.

Wireless communication describes the way to send bits into free-space through transmitter and how to capture them correctly and convert them to bits again. These days many IC chip maker have developed single chip solutions for this purpose i.e. serial data transition over a range of 100 - +200meter at a rate of 10 - 20kbps and requires only PCB (Printed Circuit Board) fabricated antenna.

Microcontroller is a wonderful piece of device. U can define it as single chip computer and comprises of a processor capable of executing microcontroller specific Assembly Language, Parallel/Serial Ports, ROM (Read Only Memory), RAM (Random Access Memory), Registers (AX, BX, CX etc.), SFR (Special Function Registers) etc. Microcontroller programmers use microcontroller specific simulators to write firmware. Simulators actually provides the software simulation of microcontroller's internal working thus helps the firmware writer to debug the firmware before actually sending the code into the microcontroller ROM using emulator.

To control high voltage (120v) equipments especially inductive loads like Fan, water pump motor etc. requires special electrical circuitry to remove transients effects that result in switching of relays connected more specifically to such loads. Such circuits basically not only provides the interface between decision making device such as microcontroller and high voltage equipments but also helps to remove surges that induce in the overall electrical circuit due to switching relays.

## SUMMARY OF THE INVENTION

Using this system, the entire home devices (Lamps, Fans etc.) can be controlled through voice. The system comprises of a single PC, transmitter and receiver module attached with each device to be controlled. The user of this system will no need to move from his/her chair to turn on/off the light bulb or fan, moreover, when user has to go out from the home, there is no need for her/him to go to each room to turn off lights, fans etc. just by speaking voice commands will do all that.

We will assign each receiver module two IDs, one will be Network\_ID and other will be Device\_ID. These IDs are assigned by system user using dip-switches on receiver module. In a particular network, every receiver module will have same Network\_ID but different Device\_IDs. After doing this, we will configure and synchronize the software having embedded voice recognition engine with these receiver modules. In the software we will assign each receiver module a unique name e.g. if we have attached one receiver module with fan as a device to be controlled then system user using the software can assign a name as "FAN" to it or whatever. That name will then be used as voice command to control that device. Under running condition, when the system user will speak "FAN" in the microphone, the software will recognize that voice command and will generate the corresponding Network\_ID and Device\_ID for that receiver module and will then build a packet of bits containing three frames as shown in Figure 4. That packet is then broadcasted through the transmitter attached with the serial port of the PC. Based on the protocol described in Figure 4A, all receiver modules will receive this packet but only the desired receiver module will respond to it.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows the hardware setup of the voice controlled wireless system. There is a single transmitter attached with the serial port of PC and receiver module is attached with each device to be controlled.

Figure 2 describes how the network would be if we want to implement the same system to entire home. Audio transmitter and microphone unit has to be installed in each room. There is a single audio receiver attached with microphone input on sound card of PC. Audio transmitter/receiver will work based on AM/FM technique whereas Digital transmitter/receiver will work based on ASK/FSK technique.

Figure 3 depicts digital receiver module schematic with electric bulb attached as a device to be controlled. Two dip-switches are shown attached with the Microcontroller marked as 1 and 2. The receiver converts the signal (ASK/FSK) from transmitter back to bits. Those bits are fed to Microcontroller serially. Switching circuit receives the signal from Microcontroller and it contains a driving circuitry to control the relay.

Figure 4A shows the protocol packet architecture used in the wireless communication between single transmitter with all receiver modules. Network\_ID is transmitted first then Device\_ID and on last Command frame.

Figure 4B describes the real world example of protocol packet where 4 bits are used for Network\_ID frame, 8 bits are used for Device\_ID frame and 4 bits are used for Command frame. In each frame, LSB (Least Significant Bit) is transmitted first.

Figure 5 tells the working of firmware inside the Microcontroller to implement protocol. Figure 5C shows the data read from 4-bit dip-switch meant for Network\_ID, is stored in BX register, with BL containing 0011.

Figure 5D shows the data read from 8-bit dip-switch meant for Device\_ID is stored in CX register of Microcontroller having value 11100011.

Figure 5A depicts that data is coming into the Microcontroller in AX register serially using scheme "Serial-in, Serial-out". Microcontroller will first look for Network\_ID and perform a comparison between AL and BL after in-taking each bit.

Figure 5B shows that after having a Network\_ID match, the Microcontroller will intake 4 bits and then perform the comparison between AX and CX

Figure 5E shows that after having Device\_ID match, the Microcontroller will intake 4 bits and check AH. If its contents are 0001 then it means, command to turn-on the attached device (e.g. Electric Bulb) and if its contents are 0000 then it means, command to turn-off the attached device.

## DETAILED DESCRIPTION OF THE INVENTION

This system consists mainly of four parts, software (runs on PC), transmitter, receiver (containing firmware in Microcontroller) and protocol. All of them are discussed one by one in the following paragraphs.

Software will run on the PC and having voice commands recognition ability and assigning each device to be controlled a unique name (e.g. Fan, Lamp etc. or whatever as desired by the system user). When the system user wants to turn on the fan, he just has to say "Fan" in the microphone attached with the PC as shown in Figure 1. Protocol has been developed and implemented in this system that describes how single transmitter will communicate with the receiver module attached with each device to be controlled. Based on this protocol, software will send three frames of bits namely, Network\_ID, Device\_ID and Command as shown in Figure 4A to the transmitter attached with the serial port of the PC.

Transmitter attached with the serial port of the PC will transmit those pulses based on the ASK (Amplitude Shift Keying) technique. Baud rate of serial port must be matched or at least less than max baud rate of the transmitter. There are chips from different brands that are single chip transmitter meant to be used with the serial port and provides a range of 100+ meters with baud rate of 10 or 20kbts/sec or even more. As in our case, we just want to turn devices on or off so having a system with baud rate of 10kbts/sec is more than enough.

Receiver module attached with each device (Fan, Lamp etc.) will receive those signals and convert them back to bits. There must be two dip-switches (4/8 bits or whatever depends upon

the number of devices to be controlled) on the receiver. One is meant for setting the Network\_ID of the receiver and other for Device\_ID.

Protocol comprises of three frames of bits as shown in Figure 4A and describes the way how software after recognizing the voice command will broadcast the command through the transmitter and how only the desired receiver will respond to that command.

Lets me first start with receiver module. The receiver module is shown in the Figure 3 with electric bulb attached. As is depicted in the Figure 3, the receiver module will contain two dip-switches. Using one we can assign Device\_ID to that receiver so that it can recognize that this command sent by transmitter is for it to execute. Other will be used to assign the Network\_ID e.g. if u want to control 12 devices using same transmitter then all receiver modules attached with each device would have same Network\_ID (same dip-switch setting) but each would be having its own unique Device\_ID (different dip-switch setting). If u want to design the system to control 256 devices then u will require 8-bits dip-switch for allotting each device a unique Device\_ID. Similarly if u have two or more such networks running in the vicinity of each other then for each receiver module to identify its own network transmitter, we will need dip-switch of more than 2 bits on the each receiver module.

Either FSK (Frequency Shift Keying) or ASK (Amplitude Shift Keying) or any other method of sending digital data can be used through transmitter. The receiver will receive those waves converts them back to bits and those bits will be read by Microcontroller through serial port.

Transmitter will first send the Network\_ID with least significant bit first. Receiver modules will first check whether it is a call from the transmitter on their network or some other network by comparing the Network\_ID set on the receiver using dip-switch with the one it received from the transmitter. If both are same then it will look for the Device\_ID and compares the one it received from the transmitter with the one system user allotted using dip-switch on the receiver. If Device\_ID comparison also results in a match then the Microcontroller will look for the Command frame. If it is 0001 then it will turn on the attached device and if it is 0000 then it will turn it off.

Now consider a network as shown in Figure 1 with one receiver module attached with the fan and other one with the electric bulb. Lets suppose we have set the dip-switch meant for Network\_ID as 0011 on both the receiver modules since both are to be controlled by same transmitter and Device\_ID for electric bulb as 11100011 and for fan as 11000011 using dip-switch meant for setting Device\_ID. Both of these dip-switches are depicted in Figure 3. After running the software on the PC, system user will first configure and synchronize the software with the transmitter and receiver modules by first entering the Network\_ID as set on each receiver module. Then system user will enter the Device\_ID of each receiver module and assign each device a name, which will become the voice command for that device e.g. if the receiver having Device\_ID of 11100011 is attached with a electric bulb then system user will write BULB (or whatever name he wants to assign to that device) in the name text box next to Device\_ID of the 11100011. In the same manner system user will enter the Device\_ID of each receiver module and assign a unique name to each device.

When the system user will activate the software and will speak BULB in the microphone, the voice recognition engine embedded in the software will recognize the voice command and software will then come to know which Device\_ID system user wants to toggle (turn On/Off). The software will then build three frames of bits as shown in Figure 4B and will send it to transmitter for broadcast. All receiver modules attached with each device will receiver those bits but only the desired receiver module will execute the command.

In the receiver module, the received signal from transmitter is converted back to bits and those bits are fed to Microcontroller serially. The working of firmware in Microcontroller is explained in Figure 5. As shown in Figure 5C, the dip-switch data meant for Network\_ID is saved in BX register with BL containing 0011 and the dip-switch data meant for Device\_ID is saved in CX register as shown in Figure 5D having value 11100011. The received data is fed serially to AX register of Microcontroller with scheme "Serial-in, Serial-out" e.g. as shown in Figure 5A, 1 will take the position of MSB (Most Significant Bit) and reset of the data in the register is shifted to the right one bit with LSB (Least Significant Bit: in this case 0) is taken out of the register.

The Microcontroller will first look for Network\_ID and will run the comparison routine after each bit in-take into the AX register. It will keep comparing the value in the AL register with



the value in the BL register as shown in Figure 5A after each bit intake, unless a match is found. If the match is found, then the Microcontroller will look for Device\_ID and will intake 4 bits the same way but without comparing the data in AX with data in CX after each bit intake. Because when AL results in a match with BL, the value in AH actually contains the 4 bits of Device\_ID frame. After 4 bits intake, the Microcontroller will then perform the comparison between AX and CX as shown in Figure 5B. If the match is found, then the Microcontroller will look for Command frame otherwise it will go back to main routine of looking for Network\_ID, as this is not the command for it.

If AX and CX results in a match as in Figure 5B, then the Microcontroller will look for Command frame and will intake 4 bits and will check the content of AH as shown in Figure 5C. If it is 0001 then its a command to turn on the attached device and if it is 0000 then its a command to turn off the attached device. Based on the information in the AH register, Microcontroller will then send the order to the switching circuit meant for driving relay as shown in Figure 3.

If we have to extend the same system for complete home then we have to use one audio transmitter/receiver (using AM/FM technique) as well, as shown in Figure 2. One microphone and audio transmitter module is installed in each room. When the system user will speak "Bedroom Fan", the audio signal is transmitted and will receive by audio receiver attached with the microphone input of PC as shown in Figure 2. The voice engine embedded in the software will then recognize this voice command and software will build the frames based on the Device\_ID related to voice command, "Bedroom Fan". Those frames of bits are broadcasted through the transmitter. Based on that protocol as shown in Figure 4A only the desired device will respond to the voice command.